

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07200378 A**(43) Date of publication of application: **04.08.95**

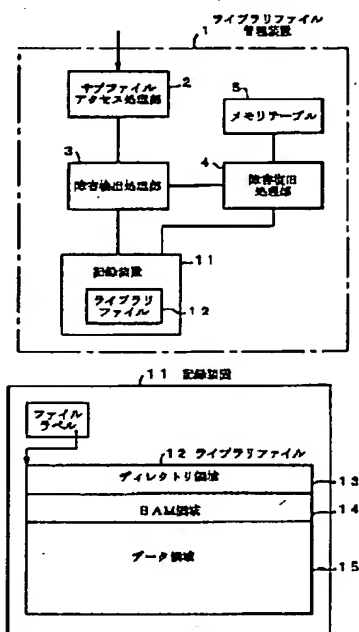
(51) Int. Cl.

G06F 12/00(21) Application number: **05351949**(71) Applicant: **NEC CORP**(22) Date of filing: **31.12.93**(72) Inventor: **SUZUKI YOSHIYUKI****(54) LIBRARY FILE MANAGEMENT EQUIPMENT****(57) Abstract:**

PURPOSE: To provide the library file management equipment in which contradiction of setting between a data area and a BAM area or contradiction of setting between the data area and a directory area is recovered even when a fault due to system-down takes place.

CONSTITUTION: The library management equipment is an equipment managing a library file 12 comprising a data area 15 to which data of a sub file are stored, a directory area 13 storing a start address and an end address of the data area, and an allocation map area 14 to which information representing the presence of the use of the data area, and the equipment is provided with a restoration means 4 which recovers contradiction of information stored in the directory area or the allocated map area based on time stamp information provided in the data area and the directory area for each sub file.

COPYRIGHT: (C)1995,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-200378

(43) 公開日 平成7年(1995)8月4日

(51) Int. Cl.⁵

G 0 6 F 12/00

識別記号

5 3 1 R

庁内整理番号

8944-5B

F I

技術表示箇所

審査請求 有 請求項の数 3 F D (全 7 頁)

(21) 出願番号 特願平5-351949

(22) 出願日 平成5年(1993)12月31日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 鈴木 至之

東京都港区芝五丁目7番1号 日本電気株式会社内

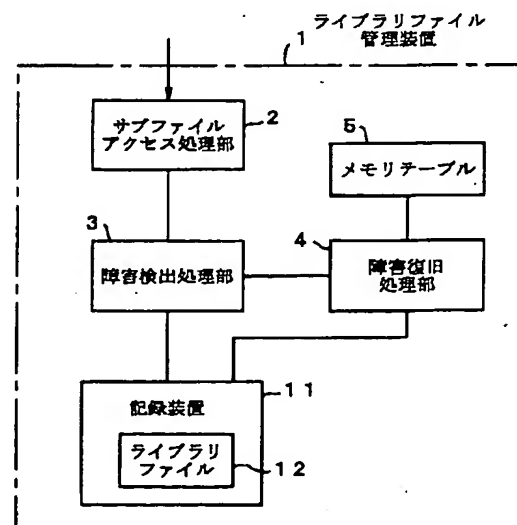
(74) 代理人 弁理士 松本 正夫

(54) 【発明の名称】 ライブラリファイル管理装置

(57) 【要約】

【目的】 システムダウン等による障害が発生した場合でも、データ領域とBAM領域の設定の矛盾あるいはディレクトリ領域の矛盾を復旧することができるライブラリファイル管理装置を提供する。

【構成】 サブファイルのデータを格納するデータ領域15と、データ領域の開始アドレスと終了アドレスを格納するディレクトリ領域13と、データ領域の使用の有無を示す情報を格納する割当マップ領域14とからなるライブラリファイル12の管理を行なうライブラリファイル管理装置であり、データ領域15及びディレクトリ領域13が、サブファイル毎のタイムスタンプ情報を備え、データ領域とディレクトリ領域のタイムスタンプ情報に基づいて、ディレクトリ領域または割当マップ領域に格納された情報の矛盾を復旧する復旧手段4を備える。



【特許請求の範囲】

【請求項1】 サブファイルのデータを格納するデータ領域と、該データ領域の開始アドレスと終了アドレスを格納するディレクトリ領域と、前記データ領域の使用の有無を示す情報を格納する割当マップ領域とからなるライブラリファイルの管理を行なうライブラリファイル管理装置において、

前記データ領域及びディレクトリ領域が、サブファイル毎のタイムスタンプ情報を備え、

前記データ領域とディレクトリ領域の前記タイムスタンプ情報に基づいて、前記ディレクトリ領域または割当マップ領域に格納された情報の矛盾を復旧する復旧手段を備えることを特徴とするライブラリファイル管理装置。

【請求項2】 前記データ領域に格納されたサブファイルのアクセスに際して、前記データ領域とディレクトリ領域のタイムスタンプ情報を比較し、不一致の場合に障害が発生したことを検知する障害検知手段を備えることを特徴とする請求項1に記載のライブラリファイル管理装置。

【請求項3】 前記復旧手段は、前記割当マップ領域の情報で使用中とされている前記データ領域のタイムスタンプ情報が、前記ディレクトリ領域に存在するか否かを判定し、存在しない場合に当該データ領域を無効として、前記割当マップ領域の情報を未使用に変更して前記割当マップ領域の矛盾を復旧し、

前記ディレクトリ領域の開始アドレスで示される前記データ領域から順次検索を行ない、前記タイムスタンプ情報が前記ディレクトリ領域と異なる前記データ領域を検出した場合、当該データ領域の1つ前のデータ領域までを有効とし、該1つ前のデータ領域のアドレスを前記ディレクトリ領域の最終アドレスとして前記ディレクトリ領域の矛盾を復旧することを特徴とする請求項1に記載のライブラリファイル管理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はライブラリファイル管理装置に関し、特に、システムダウン等で発生した障害の検知と復旧に特徴を有するライブラリファイル管理装置に関する。

【0002】

【従来の技術】 互いに独立したデータ群は、一般に別々なファイルに格納される。しかし、それらのデータ群が同一の属性を持つ場合、一つのファイルとして集中的に管理するのが、ファイルのアサイン、スペース管理及びアクセスの面で効率的であることが多い。このような集中管理を可能にするのが、ライブラリファイル管理装置であり、個々のデータ群は、サブファイルとして管理される。

【0003】 ライブラリファイルは、一般にディレクトリ領域、ブロック割当マップ（BAM）領域、データ領

域の3つの領域によって構成される。ディレクトリ領域は、ディレクトリブロックと呼ばれるブロックから構成され、サブファイルの記述情報を管理する。BAM領域は、BAMブロックと呼ばれるブロックから構成され、データ領域の空きスペースを管理する。また、データ領域は、データブロックと呼ばれるブロックから構成され、各サブファイルのデータが格納される。

【0004】 ライブラリファイルでは、サブファイルが使用するブロックの割当を論理トラック単位（ブロックの集まり）として管理している。論理トラックの割当方式としての1論理トラック単位に領域を取り出す方式では、1論理トラックが1ブロックから構成される場合、1ブロックを取り出す度にBAMブロックの入出力が発生して処理効率が大幅に低下する。

【0005】 そこで、ライブラリファイルのアクセスでは、空き論理トラックが必要となった時、一度のいくつかの空き論理トラックをBAM領域から取り出し、メモリ上の制御テーブルでプール化し、プール中の空き論理トラックがなくなった時に再度いくつかの論理トラックをBAM領域から取り出す。また、不要となった論理トラックの開放時には、プールへ返却し、プールが一杯になった時には、残りの不要な論理トラックのリストを作成し、そのリストを元にしてBAM領域への返却を行っている。

【0006】

【発明が解決しようとする課題】 上述した従来のライブラリファイル管理装置では、システムダウンやジョブアポート等の時に、データ領域とBAM領域との間で不一致が生じるおそれがあった。例えば、BAM領域より空きスペースを確保して使用中とした場合、その時点でまだデータ領域へのデータ出力が行なわれる前にシステムダウン等が発生すると、データ領域上は無効なデータであるにも関わらず、BAM領域にはデータ領域の論理トラックが使用中である旨の情報が設定されることになり、不一致が生じ、また、ディレクトリ領域の内容に矛盾が発生し、正確なアクセスが行なわれなくなる問題があった。

【0007】 本発明は、上記従来の欠点を解決し、システムダウン等による障害が発生した場合でも、障害復旧機能によって、データ領域とBAM領域の設定の矛盾あるいはディレクトリ領域の矛盾を復旧することができるライブラリファイル管理装置を提供することを目的とする。

【0008】

【課題を解決するための手段】 上記目的を達成するため、本発明は、サブファイルのデータを格納するデータ領域と、該データ領域の開始アドレスと終了アドレスを格納するディレクトリ領域と、前記データ領域の使用の有無を示す情報を格納する割当マップ領域とからなるライブラリファイルの管理を行なうライブラリファイル管

理装置において、前記データ領域及びディレクトリ領域が、サブファイル毎のタイムスタンプ情報を備え、前記データ領域とディレクトリ領域の前記タイムスタンプ情報に基づいて、前記ディレクトリ領域または割当マップ領域に格納された情報の矛盾を復旧する復旧手段を備える構成としている。

【0009】他の好ましい態様によれば、前記データ領域に格納されたサブファイルのアクセスに際して、前記データ領域とディレクトリ領域のタイムスタンプ情報を比較し、不一致の場合に障害が発生したことを検知する障害検知手段を備える構成としている。

【0010】また、他の好ましい態様によれば、前記復旧手段は、前記割当マップ領域の情報で使用中とされている前記データ領域のタイムスタンプ情報が、前記ディレクトリ領域に存在するか否かを判定し、存在しない場合に当該データ領域を無効として、前記割当マップ領域の情報を未使用に変更して前記割当マップ領域の矛盾を復旧し、前記ディレクトリ領域の開始アドレスで示される前記データ領域から順次検索を行ない、前記タイムスタンプ情報が前記ディレクトリ領域と異なる前記データ領域を検出した場合、当該データ領域の1つ前のデータ領域までを有効とし、該1つ前のデータ領域のアドレスを前記ディレクトリ領域の最終アドレスとして前記ディレクトリ領域の矛盾を復旧する構成としている。

【0011】

【実施例】次に本発明の好適な実施例について図面を参照して説明する。図1は本発明のライブラリファイル管理装置の全体構成を示すブロック図である。図1において、本ライブラリファイル管理装置は、サブファイルアクセス処理部2、障害検出処理部3、障害復旧処理部4、メモリーテーブル5、ライブラリファイル12を格納する記憶装置11から構成される。

【0012】サブファイルアクセス処理部2は、ライブラリファイル12に対するサブファイルの作成、追加、削除等の処理を行なう。また、障害検出処理部3は、ライブラリファイル12に障害が発生したことを検知する。さらに、障害復旧処理部4は、障害が発生したライブラリファイル12の内容を復旧する処理を行なう。メモリーテーブル5は、障害復旧処理に際してディレクトリブロックのタイムスタンプデータを格納する。

【0013】ライブラリファイル12を格納する記憶装置11としては、ランダムアクセスが可能な装置が使用され、例えば、磁気ディスクや光ディスクが使用される。記憶装置11には、1または複数のライブラリファイル12が格納される。また、1つのライブラリファイル12には、1つあるいは複数のサブファイルが含まれる。

【0014】図2にディスク装置11に格納されるライブラリファイル12の構成を示す。ライブラリファイル12は、ディレクトリ領域13、BAM領域14、デー

タ領域15により構成される。ディレクトリ領域13は、ディレクトリブロック21の集合により構成され、BAM領域14は、BAMブロック22の集合により構成され、データ領域15は、データブロック23の集合により構成される。

【0015】図3に上記ライブラリファイル12の各領域の構成例を示す。ディレクトリ領域13を構成するディレクトリブロック21は、管理情報21aと複数のディレクトリエントリ21bで構成される。管理情報21aには、使用バイト数を書き込む。また、ディレクトリエントリ21bは、サブファイル名25a、開始ブロックアドレス25b、終了ブロックアドレス25c、タイムスタンプ25d、その他の情報25eを有する。

【0016】サブファイル名25には、サブファイル毎に付けた名称を書き込む。開始ブロックアドレス25b及び終了ブロックアドレス25cには、サブファイルのデータを書き込んだ最初のデータブロック23のアドレス及び最後のデータブロック23のアドレスを設定する。タイムスタンプ25dには、サブファイル作成時の年月日及び時間を設定する。また、その他の情報25eには、サブファイルの識別子等を設定する。

【0017】BAM領域14を構成するBAMブロック22は、NEXTポインタ22a、PRIORポインタ22b、BAM記述22Cで構成される。NEXTポインタ22a及びPRIORポインタ22bには、次のBAMブロック22の位置及び一つ前のBAMブロック22の位置を設定する。また、BAM記述22Cには、フォーマット済みの論理トラック数24a、未使用の論理トラック数24b、各データブロック23の使用の有無を示す割当マップ24cが含まれる。割当マップ24cについては、データブロック23が使用中の場合に

「1」が、未使用の場合に「0」が設定される。

【0018】データ領域15を構成するデータブロック23は、NEXTポインタ23a、PRIORポインタ23b、その他の情報23c、タイムスタンプ23d、複数のレコード1～nで構成される。NEXTポインタ23a及びPRIORポインタ23bには、次のデータブロック23の位置及び一つ前のデータブロック23の位置を設定する。その他の情報23cには、ブロック内に含まれているレコード数、ブロック中のバイト数などを設定する。タイムスタンプ23dには、ディレクトリエントリのタイムスタンプ25dと同様、サブファイル作成時の年月日及び時間を設定する。また、各レコードには、データを書き込む。

【0019】ここで、ライブラリファイル12にサブファイルを作成する際の処理内容を図4から図6を用いて説明する。図4は、ライブラリファイル12にサブファイルをオープンするときの状態を示す。まず、ディレクトリブロック21に新規のサブファイルに対応するディレクトリエントリ21bを作成する。

【0020】そのディレクトリエントリ21bのサブファイル名25aには、ファイル名を設定し、開始ブロックアドレス25bには、最初に確保したデータブロック23のアドレスを設定し、終了ブロックアドレス25cには、「0」を設定する。また、タイムスタンプ25dには、その作成時の年月日及び時間を取得して設定する。図4では、1論理トラックに2つのデータブロック23が格納される場合を示している。

【0021】図5は、確保したデータブロック23にデータを出力する状態を示す。データは、確保したデータブロック23に順次書き込む。データブロックが不足した場合には、論理トラック単位でデータブロック23を確保して追加する。各データブロック23間については、NEXTポインタ23a、PRIORポインタ23bによって接続する。また、ディレクトリエントリ21bに設定したタイムスタンプ25dと同一の値を、データブロック23のタイムスタンプ23dに設定する。

【0022】図6は、サブファイルをクローズする時の状態を示す。全てのデータの書き込みが終了した場合、ディレクトリエントリ21bの最終ブロックアドレス25cに、最後のデータブロック23のアドレスを書き込む。

【0023】サブファイル作成時のシステムダウンによる障害では、BAMブロック22の設定上使用中になっているデータブロックにデータが未更新である場合、ディレクトリエントリ21bの最終ブロックアドレス25bの値と、最終データブロック23のアドレスとが一致していない場合が発生する。このような障害の復旧を障害復旧処理部4が行なう。

【0024】ここで、障害復旧機能を実行せずにサブファイルをアクセスした場合の、障害検知処理部3による障害検知と、障害を検知した場合の障害復旧処理部4による復旧処理について説明する。ここでは、図5のデータ出力において、データブロック23(5)にデータを書き込んだ後にシステムダウンが起こった場合を例にとって説明する。

【0025】まず、障害検知処理部3では、ディレクトリエントリ21bの終了ブロックアドレス25cが「0」か否かを判定する。「0」であれば、サブファイルクローズ前の障害と判定し、ディレクトリエントリ21bの開始ブロックアドレス25bを元に、先頭のデータブロック23(1)を検索し、NEXTポインタ23aを参照して順次データブロック23を検索する。その際、データブロック23のタイムスタンプとディレクトリエントリ21bのタイムスタンプとの比較を行なう。両者のタイムスタンプが不一致であれば、そのデータブロック23のでデータ書き込み前にシステムダウンしたことが検知される。

【0026】図5の例では、データブロック23(5)のデータ書き込みが終了して、そのNEXTポインタ2

3aは、データブロック23(6)を示している。しかし、データブロック23(6)は、データ書き込みが終了しておらず、このため、データブロック23(6)とのタイムスタンプの比較により不一致が検出されることになる。

【0027】上記のような不一致が障害検知処理部3で検知された場合、障害復旧処理部4は、一つ前のデータブロック23(5)までを有効とし、ディレクトリエントリ21bの最終ブロックアドレス25cに、データブロック23(5)のアドレスを設定し、かつデータブロック23(5)のNEXTポインタ23aには、最終ブロックであることを示す値(NUL)を設定する。

【0028】併せて、使用者に対して、BAMブロック22上は使用中であるが、データブロック23上使用されていないデータブロックが存在する可能性があるため、障害復旧機能実行の警告を行なう。

【0029】サブファイルの終わりへのレコードの追加を行なう場合は、追加処理モードで該当するサブファイルをオープンし、ディレクトリエントリ21bよりタイムスタンプ25dを取得する。ディレクトリエントリ21bで示される最終のデータブロック23に空きが存在すれば、そこにレコードを追加する。空が存在しなければ、新たにデータブロック23を確保する。そして、確保した追加データブロック23には、ディレクトリエントリ21bより取得したタイムスタンプを設定する。レコードの追加が全て終了した場合、サブファイルをクローズし、ディレクトリエントリ21bの終了ブロックアドレス25cに追加した最後のデータブロック23のアドレスを設定して更新する。

【0030】サブファイルへのレコード追加時にシステムダウンが発生した場合、上記したサブファイル作成時と同様な不一致状態が発生し得る。復旧については、上述と同様に行なう。なお、障害復旧機能を実行せずに、該当するサブファイルをアクセスした場合の障害検出においては、ディレクトリエントリ21bの最終ブロックアドレス25cにより最終のデータブロックを検索し、そのデータブロック23のNEXTポインタ23aが最終ブロックであることを示す値(NUL)以外の値であれば、追加中にシステムダウンが発生したことを検知できる。

【0031】この状態を検知した場合、NEXTポインタ23aに基づいて順次データブロックを検索し、タイムスタンプの不一致を検出する。タイムスタンプの不一致を検知した後は、上述したサブファイル作成時の処理と同様の復旧処理を行なうものである。

【0032】サブファイルの削除処理の時は、ディレクトリエントリ21bより削除対象となるサブファイルの情報をまず削除する。次に、BAMブロック22よりそのサブファイルの使用していた、全ての論理トラックを開放する。データブロック23については、I/Oの削

10

20

30

40

50

減によりオーバーヘッドを軽減するため、ブロックの初期化は行なわない。

【0033】このサブファイル削除時のシステムダウンでは、BAMブロック22上は使用中であるが、該当するサブファイルが存在しない状態が発生する。復旧については、上述したサブファイル作成時の処理と同様の復旧処理を行なう。

【0034】次に、障害復旧処理部4による障害復旧機能について、図7から図9を用いて詳細に説明する。最初に、データ領域15とBAM領域14との矛盾検査によるBAMブロック22の障害復旧を説明する。図7に示すように、ディレクトリ領域13上の全てのディレクトリブロック21を読み込む。読み込んだ内容を、タイムスタンプをキーとしてソートし、メモリテーブル5に格納する。次に、BAM領域14より先頭のBAMブロック22を読み込む。そして、BAMブロック22上で割当マップ24cに使用中として設定されているデータブロック23の内容を読み込む。BAMブロック22上で未使用のデータブロックであれば、特に読み込みを行なわない。

【0035】読み込んだデータブロック23のタイムスタンプ23dが、メモリテーブル5上に存在するかどうかを調べる。存在しない場合は、システムダウン等による無効なデータブロックであると判定し、BAMブロック22の割当マップ24cが使用中「1」となっている場合は、未使用状態「0」に設定する。順次以上の処理を、全てのデータブロック23に対して実行する。図8においては、BAMブロック22のBAM記述22cの中の割当マップ24cの値のみを取り出して示している。

【0036】図7から図9で示す例では、図8に示すように、ブロック番号1の第1のデータブロック、ブロック番号2の第2のデータブロック23はBAMブロック22上使用中となっているので、読み込みが行なわれる。タイムスタンプは、図7におけるファイルCの値と一致するので、正しいデータブロックとして判定される。第3及び第4のデータブロックは、BAMブロック22上未使用「0」となっているため、読み込まれない。

【0037】使用中となっている第5のデータブロック23は、タイムスタンプがファイルDと一致する。また、第6のデータブロック23のタイムスタンプは、メモリテーブル5上に対応する値が存在しない。このため、未使用のデータブロック23と判定され、BAMブロック22上の割当マップ24cの状態を未使用に変更する。

【0038】次に、第2の復旧処理として、ディレクトリエントリ21bの矛盾修正処理を説明する。まず、先頭のディレクトリエントリ21bを読み込む。読み込んだディレクトリエントリ21bの、開始ブロックアドレ

ス25bから該当するサブファイルの先頭データブロック23を検索する。

【0039】次に、検索したデータブロック23のNEXTポインタ23aを参照して順次データブロック23を検索し、NEXTポインタ23aが最終データブロックを示すまで検索を実施する。検索した最終データブロック23について、そのタイムスタンプの値とディレクトリブロックのタイムスタンプを比較する。比較の結果、異なれば、1つ前のデータブロック23までを有効とし、ディレクトリエントリ21bの最終ブロックアドレス25cをその1つ前のデータブロック23のアドレスを設定して更新する。この処理を全てのディレクトリエントリ21bについて実行する。以上の障害復旧処理によってライブラリファイルの復旧が完了することになる。以上好ましい実施例をあげて本発明を説明したが、本発明は必ずしも上記実施例に限定されるものではない。

【0040】

【発明の効果】以上説明したように本発明のライブラリファイル管理装置によれば、システムダウン等による障害が発生した場合でも、障害復旧機能によって、データ領域とBAM領域の設定の矛盾あるいはディレクトリ領域の矛盾を復旧することができるので、障害が発生してもサブファイルを正確にアクセスすることができる。また、障害の復旧前でも、アクセスすることで障害が発生したことを検知することができる。

【図面の簡単な説明】

【図1】 本発明のライブラリファイル管理装置の全体構成を示すブロック図である。

【図2】 本発明で管理するライブラリファイルの構成を示す図である。

【図3】 ライブラリファイルに含まれる各領域を構成するブロックの構成を示す図である。

【図4】 サブファイルの作成時のオープン状態を示す図である。

【図5】 サブファイルの作成時のデータ書き込み状態を示す図である。

【図6】 サブファイルの作成時のクローズ状態を示す図である。

【図7】 ディレクトリブロックの内容の例を示す図である。

【図8】 BAMブロックの内容例を示す図である。

【図9】 データブロックの内容例を示す図である。

【符号の説明】

- 1 ライブラリファイル管理装置
- 2 サブファイルアクセス処理部
- 3 障害検知処理部
- 4 障害復旧処理部
- 5 メモリテーブル
- 11 記憶装置

9

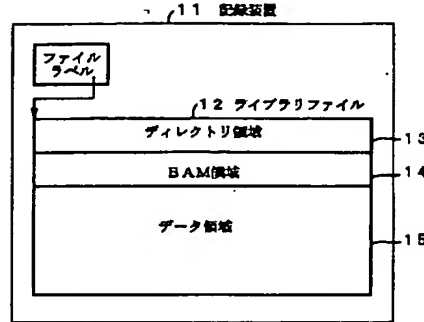
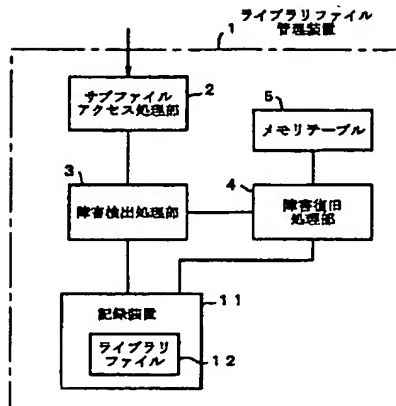
10

- 12 ライブラリファイル
13 ディレクトリ領域
14 BAM領域
15 データ領域

- 21 ディレクトリブロック
22 BAMブロック
23 データブロック

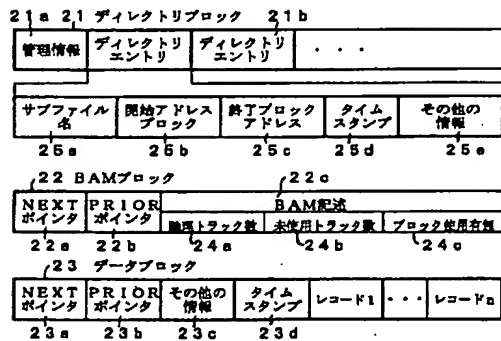
【図1】

【図2】

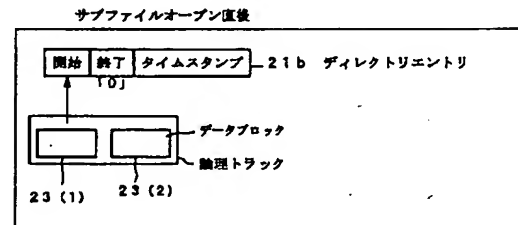
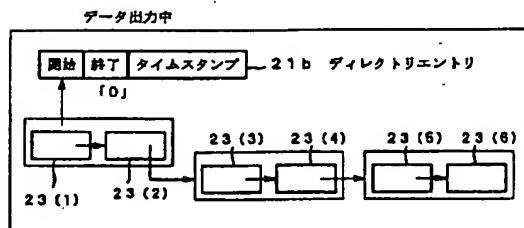


【図4】

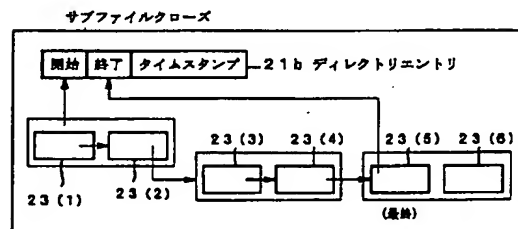
【図3】



【図5】



【図6】



【図7】

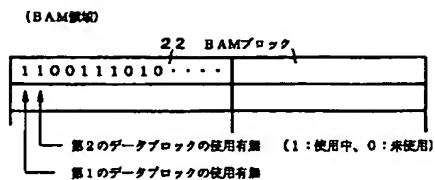
(ディレクトリ領域)

| ファイル名 | 開始 | 終了 | タイムスタンプ | メモリテーブル |
|--------|-----|-----|----------------|----------------|
| ファイル名A | 120 | 145 | 92101011300500 | 92101011300500 |
| ファイル名A | 120 | 145 | 91050516241210 | 91050516241210 |
| ファイル名A | 120 | 145 | 92091514451605 | 92091514451605 |
| ファイル名A | 120 | 145 | 93032114192008 | 93032114192008 |

21 ディレクトリブロック

読み込み
ゾート

【図8】



【図9】

(データ領域)

| ブロック 番号 | NEXT ポインタ | PRIOR ポインタ | タイムスタンプ 23 d | 23 データブロック |
|------------|--------------|---------------|-----------------|------------|
| 1 | 2 | NULL | 92091514451605 | |
| 2 | 5 9 | 2 | 92091514451605 | |
| 3 | 5 9 | 2 | 90090810101521 | |
| 4 | 5 9 | 2 | 90100812421205 | |
| 5 | 5 9 | 2 | 93032114192008 | |
| 6 | 5 9 | 2 | 89070712001510 | |
| 7 | 5 9 | 2 | 93032114192008 | |
| 8 | 5 9 | 2 | 90021519200825 | |
| | | | | |